Ügyfélszolgálati beszélgetések nyelvmodellezése rekurrens neurális hálózatokkal

Tarján Balázs[1,3], Fegyó Tibor[1,3], Mihajlik Péter[1,2]

[1] Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék
tarjanb@tmit.bme.hu
[2] THINKTech Kutatási Központ Nonprofit Kft.
mihajlik@thinktech.hu
[3] SpeechTex Kft.
tfegyo@speechtex.com

Kivonat: A spontán, társalgási beszéd leírása a mai napig komoly kihívás elé állítja a gépi beszédfelismerő rendszereket. A témák sokszínűsége és a kevés tanítóadat különösen megnehezíti a nyelvi modellek tanítását. Cikkünkben telefonos ügyfélszolgálati beszélgetéseket modellezük rekurrens LSTM neurális hálózat segítségével, mellyel közel felére sikerült csökkentenünk a perplexitást a hagyományos, count n-gram modellhez képest. Azt találtuk, hogy a rekurrens LSTM akkor is felülmúlja a count modell pontosságát, ha memóriája hosszát alacsonyra korlátozzuk (LSTM n-gram). 10 vagy annál nagyobb fokszámú LSTM n-grammal pedig a korlátozás nélküli LSTM nyelvi modell teljesítménye is megközelíthető. Ez alapján arra következtetünk, hogy a rekurrens neurális nyelvi modellek pontosságának titka a hatékony simításban rejlik, nem a hosszú távú memóriában. Az új, neurális nyelvmodell segítségével nem csak a perplexitást sikerült csökkentenünk, hanem a kapcsolódó beszédfelismerési feladaton a szóhiba-arányt is relatív 4%-kal.

1 Bevezetés

A statisztikai nyelvmodellek számos természetes nyelvfeldolgozási feladatban játszanak kulcsszerepet. Nem kivétel ez alól a gépi beszédfelismerés sem, ahol hosszú időn át a szó n-gram count statisztikák alapján, maximum likelihood becsléssel tanított ún. **count n-gram** nyelvi modellek [1] voltak az egyeduralkodók. Az utóbbi évek során azonban először az előrecsatolt neurális hálózatokra épülő [2], majd a rekurrens neurális nyelvmodellek [3] megtörték ezt a dominanciát. A rekurrens modellek felépítésükből fakadóan jól modellezik a szövegben található hosszú távú függőségeket, mely képességet elsősorban Long Short-Term Memory (LSTM) [4] egységek alkalmazásával sikerült kiaknázni [5].

Cikkünkben egy kísérletsorozat első állomását mutatjuk be, melynek keretében a neurális nyelvmodellek gyakorlati alkalmazhatóságát kívánjuk feltérképezni magyar, majd terveink szerint idegen nyelvű beszédfelismerési feladatokban. Az itt bemutatott első kísérleti eredmények egy telefonos ügyfélszolgálati beszélgetések kézi leiratait tartalmazó adatbázison születtek. Azért erre az adatbázisra esett a választásunk, mert

aránylag kis mérete gyors tanítást tesz lehetővé, miközben a beszélgetések spontán jellege, illetve a kevés tanítóadat kellően megnehezíti a hagyományos count n-gram modellek dolgát.

A **rekurrens LSTM** nyelvmodellekkel bár valóban nagyon alacsony perplexitás érhető el, valós időben működő beszédfelismerő rendszerbe nehéz integrálni őket. A gyors dekódolás egyik feltétele ugyanis, hogy kellően kis méretűre tudjuk csökkenti a keresési teret, melyet megakadályoz, hogy a rekurrens LSTM nyelvi modell rengeteg belső állapotot vehet fel. A probléma megoldására született az **LSTM n-gram**ok koncepciója [6], melyben a count modellekhez hasonlóan korlátozzuk a valószínűségbecslés során figyelembe vett korábbi szavak számát. Az LSTM n-gramok angol és német nyelven is sikeresnek bizonyultak [6, 7], ezért úgy döntöttük, hogy a hagyományos LSTM nyelvmodell mellett ezt az új struktúrát is kiértékeljük és összevetjük a count modellek teljesítményével.

Fontosnak tartottuk, hogy már a kísérletsorozatunk elején szülessenek beszédfelismerési eredmények is rekurrens LSTM nyelvmodell felhasználásával. A rekurrens modellben tárolt tudás kezdeti kinyerésére egy egyszerű megoldást alkalmaztunk [8]: nagy mennyiségű szöveget generáltunk a neurális modell segítségével, melyből aztán count n-gram modellt tanítottunk és interpoláltuk az eredeti nyelvi modellel. Legjobb tudomásunk szerint ezek az első, publikált, magyar nyelvű beszédfelismerési eredmények, melyek neurális nyelvmodell felhasználásával jöttek létre.

A következő fejezetben a kísérleteinkhez használt tanító- és tesztadatbázisokat, majd utána a cikkünk fő témáját képező nyelvmodellezési módszereket mutatjuk be. A negyedik fejezetben ismertetjük a különböző eljárásokkal kapott szöveges és beszédfelismerési eredményeket, majd az utolsó fejezetben összefoglalását adjuk vizsgálataink legfontosabb eredményeinek.

2 Tanító- és tesztadatbázisok

2.1 Tanító-adatbázisok

A nyelvi modellek tanításához magyar nyelvű, telefonos ügyfélszolgálati beszélgetések anonimizált kézi leiratait tartalmazó adatbázist használtunk, melyre a továbbiakban **MTUBA** (Magyar Telefonos Ügyfélszolgálati Beszédadatbázis) néven fogunk hivatkozni. A normalizálás során eltávolítottuk az egyértelmű akusztikai megfeleltetéssel nem rendelkező tokeneket (írásjelek). Megtartottuk azonban a kézi leiratok eredeti mondathatárait, mely a feladat párbeszédes, spontán jellegéből fakadóan a szokásosnál rövidebb, átlagosan 6,9 szót tartalmazó mondatokat eredményezett.

A tanítókorpusz 290 órányi felvétel kézi leiratát, összesen **3,4 millió tokent** és **100 ezer egyedi szóalakot** tartalmazott. A tanítás gyorsítása és szótáron kívüli szavak modellezése céljából a végleges tanítószövegben csak a **leggyakoribb 50 ezer szóalak**ot tartottuk meg, a többit szótáron kívüli szóként modelleztük és <unk> szimbólummal helyettesítettük.

2.2 Tesztadatbázisok

A kísérleteinkben szereplő nyelvi modellek teszteléséhez az MTUBA e célokra kijelölt részét használtuk. A tesztszövegekben csak a tanítószöveg szűkített szótárában szereplő szóalakokat tartottuk meg, az egyéb szavakat a szótáron kívüli szavak szimbólumával (<unk>) helyettesítettük. A tesztadatbázisok vonatkozó statisztikákat a **1. táblázat** tartalmazza.

A tesztadatadatbázist két független részre bontottuk. Az első ún. **validációs tesztszöveget** a modellek hiperparaméter-optimalizálása során használtuk (learning rate szabályozás, early stopping), míg a második ún. **kiértékelő tesztadatbázist** a kész modellek szöveges és beszédfelismerési kiértékelésére. A kiértékelő tesztadatbázist további részekre osztottuk (lásd 1. táblázat). A sztereo módon rögzített felvételeken külön tudtuk vizsgálni az ügyfélszolgálatos (MTUBA sztereo 1) és az ügyfél (MTUBA sztereo 2) oldalt. Ezzel szemben a kiértékelő tesztadatbázis mono felvételein a két oldal hanganyaga egy sávra lett keverve, így csak egybe tudjuk őket kezelni (MTUBA mono).

	Validációs teszt	**Kiértékelő teszt**			
	Σ	MTUBA sztereo 1	MTUBA sztereo 2	MTUBA mono	Σ
Tokenek száma	45773	10599	4792	50921	66312
Tesztfelvétel hossza [perc]	-	127	127	478	732
OOV arány [%]	2,7	1,4	1,5	2,8	2,5

1. táblázat. A tesztadatbázisok jellemzői
(OOV (Out of Vocabulary) arány: szótáron kívüli szavak aránya)

3 Nyelvi modellezés

Cikkünk célja, hogy különböző típusú nyelvmodellezési módszereket összehasonlítsunk egy valós életből származó beszédfelismerési feladaton. Ennek érdekében hagyományos count-alapú és neurális nyelvi modelleket is alkalmaztunk. A tanítási folyamatot és az alkalmazott módszereket mutatjuk be ebben a fejezetben.

3.1 Count n-gram nyelvi modell

A hagyományos, count-alapú, Kneser-Ney eljárással simított [9] nyelvi modelleket az SRI nyelvi modellező eszköz segítségével [10] tanítottuk. Az SRI toolkit jellemzője, hogy alapértelmezésben a 3 és annál nagyobb fokszámú, csak egyszer előforduló n-gram-okat nem veszi figyelembe a tanítás során. Kísérleteinkben ezt a funkciót kikapcsoltuk, így a szokásosnál jóval több n-gramot tartalmazó nyelvi modelleket jöttek

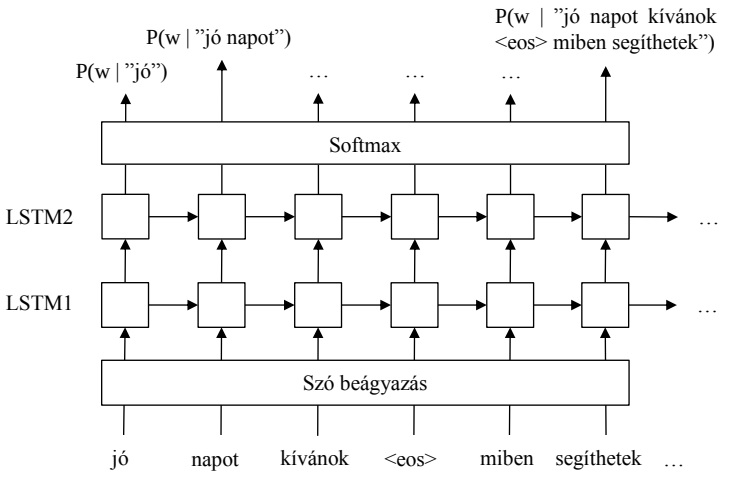
létre. Célunk ugyanis az volt, hogy egy adott fokszám mellett mindig a lehető legpontosabb count n-gram modellt tanítsuk.

Kétféle módon modelleztük a sortörést. Az első a hagyományos ún. **mondatonkénti** modellezés, melynél a sor elejére egy mondatkezdő (<s>), végére pedig egy mondatzáró (</s>) szimbólumot helyezünk, és nem engedjük meg, hogy az így létrejött mondatokon átíveljenek az n-gramok. A második, ún. **mondatösszefűzéses** módszer esetén az n-gram statisztika összeállítása során megengedjük a sorok között átívelő n-gramokat. A mondathatárok visszaállíthatósága érdekében azonban ennél a módszernél is jelöljük a sorok végét, melyre egy normál szóként modellezett speciális szimbólum szolgál (<eos>).

3.2 LSTM nyelvi modell

Az egyik nyelvi modell típus, mellyel a count n-gram nyelvi modelleket összevetjük cikkünkben egy 2 rétegű, Long Short-Term Memory (LSTM) [4] egységet tartalmazó, rekurrens neurális hálózat. Ezzel a típusú hálózattal korábban sikerült jelentős perplexitás csökkenést elérni a Penn Tree Bank (PTB) adatbázison [6, 11]. A hálózat felépítését az **1. ábra** szemlélteti.

A szavakat először egy szóbeágyazó mátrix segítségével vektorrá alakítjuk. A tanítás során ezután a szóvektorokat átvezetjük egy dropout [12] rétegen. A szóvektorok innen az első LSTM rétegre kerülnek, melynek kimenete egy dropout rétegen keresztül a következő LSTM réteg bemenetére van kötve. A második LSTM réteg kimenete egy újabb dropout réteg alkalmazása után kerül a softmax rétegre, melynek mérete megegyezett az alkalmazott szótár méretével. A softmax kimenetén a következő szóra vonatkozó valószínűségi eloszlást kapjuk, melyet úgy érünk el, hogy a tanítás folyamán mindig a következő szó a target, melyhez képest a hibát mérjük (cross entropy).



1. ábra: A kísérleteink során használt rekurrens LSTM nyelvi modell struktúra
(P(w | ”history”), a history után becsült szóeloszlást jelöli)

Keras [13] implementációnk[1] alapjául a Tensorflow minta nyelvi modell kódja szolgált[2], mely a [11]-ban bemutatott PTB modelleket valósítja meg. A fenti megvalósításból a hiperparaméterek egy részét is átvettük, így 650 dimenziós szóvektorokat, 650 dimenziós kimeneti réteggel rendelkező LSTM-eket, 35 hosszú szekvenciákat és 0,5-ös eldobási rátájú dropout-ot alkalmaztunk. Több optimalizáló függvény kipróbálása után végül a momentummal gyorsított, hagyományos Stochastic Gradient Descent (SGD) mellett döntöttünk. A tanulási ráta kezdeti értékének 1-et állítottunk be, és minden epoch végén feleztük, amennyiben hibanövekedést tapasztaltunk a validációs teszthalmazon. A tanítást akkor fejeztük be, ha három egymást követő epoch után sem regisztráltunk javulást a validációs teszten.

A **mondatösszefűzéses** LSTM nyelvi modellnél az egyes batch-ek között megőrizzük az LSTM állapotokat, azaz Tensorflow terminológiával élve „stateful" hálózatot tanítunk. **Mondatonkénti** modelleknél a sorok végén töröltük az LSTM állapotokat. A batch mérete mindkét modelltípusnál 32 volt. Többféle előre betanított szóbeágyazó mátrixszal [14, 15] próbáltuk modelljeink pontosságát javítani, de legtöbb esetben semekkora, vagy csak marginális mértékű javulást tapasztaltunk.

3.3 LSTM n-gram nyelvi modell

A hagyományos LSTM nyelvi modellek egyik hátránya, hogy a „stateful" felépítésből fakadóan a tanítás során nem lehet a tanítómintákat véletlenszerűen keverni, hanem azoknak mindig előre meghatározott sorrendben kell érkezniük. Ezen felül a batch méret növelése sincs jó hatással a pontosságukra, mivel csökken az egy egységként modellezett szöveghossz. A nehézkes tanítás mellett a gyakorlatban az is gondot okoz, hogy az LSTM modellek nagyon sok belső állapotot vehetnek fel ($H \in \mathbb{R}^h$, ahol h az LSTM réteg mérete).

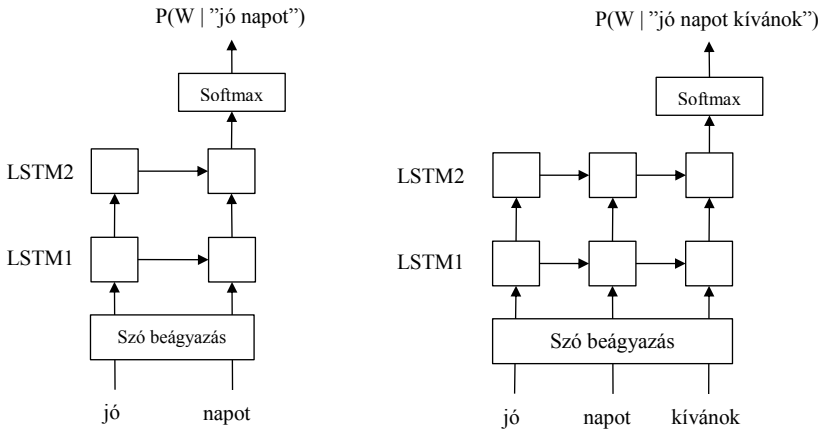
A fenti hátrányok kiküszöbölése érdekében merült fel, hogy érdemes lehet a becsléshez használt mintákat n-gramokba szervezik, és így tanítani rekurrens LSTM nyelvi modelleket [6]. Az ún. **LSTM n-gramok** használatakor tehát a count n-gramokhoz hasonlóan korlátozzuk a becslés során figyelembe vett előtörténet hosszát. Első közelítésben ez egy meglepő döntés, hiszen a rekurrens LSTM hálózatok előnyének tradicionálisan a hosszú távú függőségek jó modellezését tekintik, azonban angol és német nyelveken végzett korábbi vizsgálatok kimutatták, hogy a modellezés pontossága nem feltétlenül csökken drasztikus mértékben [6, 7]. Egy 3- és egy 4-gram rekurrens LSTM nyelvi modell felépítését szemléltetjük a **2. ábrán**.

Az LSTM n-gram implementációnkban a folyamat a hagyományos LSTM nyelvi modellhez hasonlóan a szavak vektorizálásával kezdődik. Ezután egy dropout rétegre kerülnek a szóvektorok, ahonnan az út az első, majd a második rekurrens LSTM rétegre vezet. Az LSTM n-gram modellek esetén nem alkalmaztunk dropout-ot a két LSTM réteg között. Fontos különbség a hagyományos LSTM modellhez képest, hogy az LSTM n-gram-nál csak a teljes előtörténet beolvasása után adunk becslést az azt követő szavak eloszlására.

[1] https://github.com/btarjan/stateful-LSTM-LM

[2] https://www.tensorflow.org/tutorials/sequences/recurrent

Az LSTM n-gramok tanítása során alkalmazott hiperparaméterek és optimalizálás megegyezett a hagyományos LSTM nyelvi modell bemutatása során ismertetett értékekkel. A kivételt két paraméter képezi: a szekvencia hossza, mely természetesen illeszkedik az aktuális n-gram fokszám értékéhez (n-1), másrészt a batch mérete, melynek optimális értékét 512-ben állapítottuk meg. A **mondatösszefűzéses** és **mondatonkénti** modellek itt csak annyiban térnek el, hogy előbbieknél az n-gramok a sorokon átívelhetnek, míg az utóbbinál a sor végével lezáródnak.



2. ábra: Két példa (3- és 4-gram) a kísérleteink során használt rekurrens LSTM n-gram nyelvi modellek felépítésére

4 Eredmények

A következőkben a telefonos ügyfélszolgálati beszélgetések korpusza alapján, különböző módszerekkel tanított nyelvi modellek kiértékelését ismertetjük. Az előző fejezetben bemutatott három modelltípust vetjük össze: a klasszikus count n-gram-okat, az új state-of-the-art LSTM modelleket, illetve ez utóbbiak a gyakorlatban jobban használható, egyszerűsített változatát, az LSTM n-gramokat. A modellek kiértékelését a 2.2 fejezetben ismertetett, független teszthalmazon hajtottuk végre.

4.1 Nyelvi modellek szöveges kiértékelése

A nyelvi modellek szöveges kiértékelése során a tesztanyaghoz való illeszkedésük mértékét vizsgáltuk **perplexitás** (PPL) segítségével. A 3. fejezetben ismertetett count és LSTM n-gram nyelvi modellek mondatonkénti és mondatösszefűzéses változatainak perplexitását a fokszám növekedésének függvényében vizsgáltuk. A validációs és kiértékelő tesztszövegen kapott eredményeket a **2. táblázatban** foglaltuk össze.

A táblázat alapján az első, ami szembetűnő, hogy a mondatösszefűzéses modellek lényegesen jobban teljesítettek, mint a mondatonkénti modellek. Ez az eredmény azért is lehet elsőre meglepő, mert [6]-ban a PTB korpuszon vizsgálva nem volt szignifikáns különbség a két modellezés között. A fenti cikkel ellentétben azonban itt egy rövid sorokból álló (átlagosan kb. 7 szó soronként, míg PTB kb. 21 szó soronként), spontán beszélgetések leiratai alapján készült korpuszt vizsgáltunk, mely egyben azt is jelenti, hogy nagyobb a sorok közötti összefüggés.

	Validációs teszt				**Kiértékelő teszt**			
	Mondatonkénti		Mondatösszefűzés		Mondatonkénti		Mondatösszefűzés	
Fokszám	Count	LSTM	Count	LSTM	Count	LSTM	Count	LSTM
2	154,7	148,5	159,5	148,6	128,4	124,1	131,8	124,4
3	123,6	107,9	121,0	99,6	97,3	86,4	93,1	77,8
4	119,4	96,5	114,8	83,8	92,8	75,4	86,3	64,2
5	118,7	91,3	113,6	76,2	92,1	71,7	85,0	58,3
6	118,6	88,4	113,2	72,5	91,9	69,4	84,7	54,9
8	-	86,7	-	69,3	-	68,1	-	52,4
10	-	86,6	-	65,8	-	67,3	-	49,5
12	-	85,7	-	63,9	-	67,0	-	48,0
14	-	86,0	-	62,8	-	67,3	-	47,1
∞	-	**76,9**	-	**60,1**	-	**61,2**	-	**44,6**

2. táblázat. Nyelvi modellek perplexitása az n-gram fokszám függvényében

Érdekes továbbá megfigyelni, hogy a rekurrens neurális hálózatok perplexitása milyen sokáig mutat csökkenést a fokszám növekedésével, míg a count n-gram modell hamar telítődik. Amíg egyetlen korábbi szó alapján becsüljük a következő valószínűségét (n=2), addig nincs jelentős különbség a count és LSTM modell között. Amint azonban több szót is figyelembe veszünk a becsléshez (n>2), jelentős előnyre tesz szert a rekurrens modell. Ez azzal magyarázható, hogy a neurális modell a szóvektorok és a visszacsatolt felépítés segítségével sokkal pontosabb becslést tud nyújtani a tanítás során meg nem figyelt n-gramokra is.

A táblázatban ∞-nel jelölt sorban találhatóak a hagyományos LSTM eredmények. Mondatösszefűzéses modellezés esetén ez arra utal, hogy minden korábbi szót figyelembe veszünk a valószínűség becsléséhez. Mondatonkénti modellezés esetén a mondat hossza természetesen korlátozza a figyelembe vett szavak számát. Mondatösszefűzéses modellezésnél a fokszám növekedésével egyre közelebb kerülünk a hagyományos LSTM perplexitásához. 10-gram fölött a különbség 10% alá csökken, melyből arra következtethetünk, hogy az LSTM nyelvi modellek legfőbb előnye nem az, hogy nagyon hosszú függőségeket képesek modellezni, hanem hogy jobb általánosító képességekkel bírnak, mint a count modellek, így azoknál lényegesebben robusztusabb becslést szolgáltatnak.

4.2 Beszédfelismerési kísérletek

Kísérletsorozatunk végső célja, hogy a neurális nyelvi modellek segítségével pontosabb gépi beszédleiratozás váljék lehetővé magyar nyelven is. Ezért döntöttünk úgy, hogy a szöveges kiértékelésen túl beszédfelismerési kísérleteket is végzünk. A rekurrens neurális nyelvi modellek alkalmazása azonban nem triviális a beszédfelismerésben.

4.2.1 Neurális nyelvmodell mintavételezése

Leggyakrabban úgy hasznosítjuk a neurális nyelvi modelleket, hogy a beszédfelismerés első köre során egy count n-gram nyelvi modellel ún. lattice-t hozunk létre a felismerési hipotézisekből, majd egy második körben újra súlyozzuk a felismerési hipotéziseket a lattice-ben immáron a neurális modell segítségével. Ez a kétkörös futtatás azonban időigényes, így nem támogatja a valós idejű beszédátírást. Cikkünkben ezért egy másik, [8]-ban ismertetett módszert alkalmaztunk. Ennek lényege, hogy a betanított neurális nyelvmodell felhasználásával szöveget generálunk, melyből hagyományos count n-gram modellt tanítunk, amit utána interpolálunk az eredeti tanítószöveg modelljével. A módszer mögött az a logika, hogy ha kellően sok szöveget generálunk a neurális modellel, akkor az a szöveg jól fogja reprezentálni a neurális modell által megtanult szókapcsolati eloszlásokat.

A lehető legjobb eredmény elérése érdekében a szöveggeneráláshoz új, rekurrens LSTM nyelvmodellt tanítottunk, melynek megemeltük a szótárméretét 50000-ről 95000-re. Ennek hatására lényegesen lassabb lett a modell tanítása, de 2,5%-ról 1,9%-ra tudtuk csökkenteni az kiértékelő halmazon mért OOV arányt. Az új LSTM modell segítségével generáltunk egy közel 125 millió szavas tanítókorpuszt. Mivel a count n-gram modellek teljesítménye 4-es fokszám fölött már nem javult érdemben, 4-gram modellt tanítottunk a generált szövegből, melyet utána az eredeti tanítószöveg 4-gram modelljével (KM-2 a 3. táblázatban) interpoláltunk egy a validációs halmazon meghatározott súlyozás alapján. A kapott interpolált, 4-gram count modell nagyon nagy méretűnek adódott (100 millió n-gram), így entrópia-alapú metszés [16] segítségével négy lépésben csökkentettük a méretét (BM-4, BM-3, BM-2, BM-1).

4.2.2 Beszédfelismerési eredmények

A tesztfelvételeket a nyelvi modellek és a VoXerver nevű, WFST-alapú beszédfelismerő dekóder [17] segítségével szöveggé alakítottuk. A dekódolás során HMM-DNN hibrid megközelítésben egy három rejtett rétegű, rétegenként 2500 neuront tartalmazó, 4907 kimeneti állapottal rendelkező, előrecsatolt, mély neuronhálót alkalmaztunk. Összesen 290 órányi 8 kHz mintavételi frekvenciájú telefonos beszélgetésen tanítottuk az akusztikus modellt a KALDI toolkit [18] segítségével. Az akusztikus jellemzővektorok 13 dimenziós MFCC paraméterekre épültek, melyet LDA és MLLT lineáris transzformáció követett. Osztott, három állapotú, környezetfüggő beszédhangmodelleket használtunk. A kiértékelő teszthalmazon mért szóhiba-arányokat az **3. táblázatban** ismertetjük.

A 3. táblázatban kezdeti modellként (KM) az eredeti tanítószöveg alapján tanított 3-gram (KM-1) és 4-gram (KM-2) modellre hivatkozunk. Látható, hogy hiába növeljük a modell fokszámát 3-ról 4-re, a szóhiba-arány csak minimális mértékben csök-

ken, míg a modell mérete drasztikusan megnő. Ez a tipikus esete annak, hogy hiába tanul a modell rengeteg hosszabb n-gramot a tanítószövegből, azoknak csak kis százaléka hasznosul a tesztelés során (alacsony hit rate).

Ezzel szemben a rekurrens LSTM nyelvi modell alapján tanult n-gramok sokkal jobban hasznosíthatóak. A BM-1-es modell például méretében nagyjából megegyezik a KM-1 modellel mégis relatív 2%-kal jobb szóhiba-aránnyal rendelkezik. Ha nagyobb modellméretet is megengedünk, tovább tudjuk csökkenteni a hibát. A KM-2-vel nagyjából megegyező méretű BM-3 modell relatív 3%-kal csökkenti a szóhiba-arányt míg, ha 3GB-os memóriafoglalás is megengedett, akkor összesen 4%-os relatív szóhiba-arány csökkenést mérhetünk.

A fenti hibaarány csökkenések bizakodásra adnak okot, de természetesen messze nem tekinthetjük a problémát megoldottnak. A generált szöveg alapján történő count n-gram mintavételezéssel 90 körüli értékre sikerül csökkentenünk a nyelvi modell perplexitását (BM-4). A generáláshoz használt eredeti rekurrens LSTM nyelvi modellel azonban 56-os perplexitást mértünk a kiértékelő tesztanyagon, így látható, hogy maradt még bőven lehetőség a beszédfelismerő nyelvi modelljét javítani.

Modell	**n-gramok száma [millió]**	**Modell mérete [GB]**	**PPL [-]**	**Szóhiba-arány [%]**			
			Σ	MTUBA sztereo 1	MTUBA sztereo 2	MTUBA mono	Σ
KM-1	1,2	0,2	110,2	10,7	33,0	32,8	**29,3**
KM-2	5,0	1,3	103,0	10,5	33,3	32,7	**29,2**
BM-1	1,2	0,3	100,8	10,5	32,4	32,2	**28,7**
BM-2	4,2	0,9	93,0	10,3	31,6	31,7	**28,3**
BM-3	8,8	1,6	91,4	10,1	31,7	31,7	**28,3**
BM-4	18,5	3,1	90,5	10,0	31,8	31,5	**28,1**

3. táblázat. Beszédfelismerési eredmények a kiértékelő teszthalmazon
(KM: kezdeti modell, BM: bővített modell)

5 Összefoglalás

Cikkünk egy kísérletsorozat első állomása, melyben a neurális nyelvi modellek alkalmazását vizsgáljuk beszédfelismerő rendszerben. Kíséreteinkben egy telefonos ügyfélszolgálati beszélgetéseket tartalmazó adatbázison tanítottunk hagyományos count n-gram és rekurrens LSTM neurális nyelvi modelleket. Az LSTM nyelvi modell segítségével **közel felére tudtuk csökkenteni a kiértékelő szövegen a perplexitást**. Az LSTM nyelvi modellnek egy gyakorlatban jobban alkalmazható változata az ún. LSTM n-gram, ahol n-gramok alapján tanítjuk a rekurrens LSTM modellt, így egyben korlátozzuk a becsléshez felhasználható korábbi szavak számát.

Az LSTM n-gramok minden fokszám mellett jobbnak bizonyultak, mint a count n-gram modellek. Sőt a korlátozás nélküli LSTM modell teljesítményét is megközelítik aránylag kis fokszám mellett. Mindez arra utal, hogy a rekurrens LSTM nyelvi modellek elsősorban **a fejlettebb simítás és nem a hosszú távú memóriájuk miatt pontosabbak**, mint a count n-gram modellek.

Az LSTM nyelvi modellek szöveges kiértékelésén túl arra is kísérletet tettünk, hogy a beszédfelismerés minőségét is javítsuk vele. Erre jelenlegi cikkünkben egy egyszerű módszer vetettünk be: nagy mennyiségű szöveget generáltunk a neurális modellel, majd az ebből tanított count n-gram modellt adaptáltuk az eredeti n-gram modellel. Az eredmények azt igazolják, hogy a generált szöveg hasznos n-gramokat tartalmaz, mivel belőle épített count modellel relatív **4%-kal sikerült csökkentenünk a kiértékelő teszt szóhiba-arányát**. A 4%-os hibacsökkenést eredményező bővített count modell perplexitása 90 volt, ami ha figyelembe vesszük, hogy a kezdeti modell perplexitása 103, míg a generálást végző LSTM modell perplexitása 56 volt, azt jelenti, hogy a potenciális perplexitás javulásnak a 28%-át sikerült egyelőre a beszédfelismerő rendszerben is hasznosítani.

A jövőbeli terveink között szerepel a kísérleteink kiterjesztése más, nagyobb méretű adatbázisokra. Ezen kívül újabb nyelvi modellezési technikákat és céljainak jobban megfelelő szóbeágyazási modelleket is ki kívánunk próbálni. Nyelvünk gazdag morfológiáját közvetlenül nem modellezik a jelenlegi módszerek, melyen a jövőben szintén változtatni szeretnénk.

Köszönetnyilvánítás

Kutatásunk az EUREKA_15-1-2016-0019 azonosító számú DANSPLAT projekt támogatásával készült.

Bibliográfia

1. Jelinek, F., Mercer, R.I.: Interpolated estimation of Markov source parameters from sparse data. In: Pattern recognition in practice. Proc. workshop Amsterdam, May 1980. p. 381–397,401 (1980).
2. Arisoy, E., Chen, S.F., Ramabhadran, B., Sethy, A.: Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition. IEEE Trans. Audio, Speech Lang. Process. 22, 184–192 (2014).
3. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association. pp. 1045–1048 (2010).
4. Hochreiter, S., Schmidhuber, J.J.: Long short-term memory. Neural Comput. 9, 1–32 (1997).
5. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association (2012).
6. Chelba, C., Norouzi, M., Bengio, S.: N-gram Language Modeling using Recurrent Neural Network Estimation. CoRR. 1703.10724, (2017).

7. Tüske, Z., Schlüter, R., Ney, H.: Investigation on LSTM Recurrent N-gram Language Models for Speech Recognition. In: Interspeech 2018. pp. 3358–3362. ISCA, ISCA (2018).
8. Deoras, A., Mikolov, T., Kombrink, S., Karafiát, M., Khudanpur, S.: Variational approximation of long-span language models for LVCSR. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. pp. 5532–5535 (2011).
9. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. Comput. Speech Lang. 13, 359–393 (1999).
10. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Proceedings International Conference on Spoken Language Processing. pp. 901–904. , Denver, US (2002).
11. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent Neural Network Regularization. CoRR. 1409.2329, (2014).
12. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J. Mach. Learn. Res. 15, 1929–1958 (2014).
13. Chollet, F., others: Keras, (2015).
14. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Trans. Assoc. Comput. Linguist. 5, 135–146 (2017).
15. Makrai, M.: Filtering Wiktionary Triangles by Linear Mapping between Distributed Word Models. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). pp. 2766–2770. European Language esources Association (ELRA), Portorož, Slovenia (2016).
16. Stolcke, A.: Entropy-based pruning of backoff language models. In: Proc. of the DARPA Broadcast News Transcription and Understanding Workshop. pp. 270–274 (2000).
17. Tarján, B., Mihajlik, P., Balog, A., Fegyó, T.: Evaluation of lexical models for Hungarian Broadcast speech transcription and spoken term detection. In: 2nd International Conference on Cognitive Infocommunications (CogInfoCom). pp. 1–5. , Budapest, Hungary (2011).
18. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: IEEE Workshop on Automatic Speech Recognition and Understanding. pp. 1–4 (2011).